00-335
1496.00154

INTEGRATION OF VIDEO PROCESSING INTO A BLOCK MOVE ENGINE

Cross Reference to Related Applications

The present application may relate to co-pending application Serial No. 09/900,940 filed July 9, 2001, Serial No. 09/882,971, filed June 15, 2001, Serial No. 09/878,594 filed June 11, 2001, Serial No. 09/___,___ filed concurrently (Attorney Docket 1496.00119) and Serial No. 09___,___ filed concurrently (Attorney Docket 1496.00165) which are each hereby incorporated by reference in their entirety.

Field of the Invention

The present invention relates to a method and/or architecture for video processing generally, and more particularly to a method and/or architecture for integrating video and graphics processing in a single processor.

Background of the Invention

The implementation of a block move engine (BME) (a bit blitter or blitting engine) for rapidly copying blocks of graphics data from one location in memory to another is generally used for

graphics processing. BMEs may be extended to include two input data streams of identical size, which are combined by a logical composition operation and written back to memory as a single data block. The demand for improvements in graphics speed and resolution and the convergence of video and graphics applications onto common platforms has made it is desirable to incorporate a wider selection of functions within the general structure of a BME.

Summary of the Invention

The present invention provides an apparatus comprising a data modification circuit and a composite circuit. The data modification circuit may be configured to generate a first output data stream in response to a first one or more of the data streams. The composite circuit may be configured to generate a combined output data stream in response to the first data stream and remaining data streams.

The objects, features and advantages of the present invention include providing a method and/or architecture for implementing a block move engine (BME) that may (i) read any graphics or video objects, (ii) manipulate and combine the objects and/or (iii) write the results into a memory.

Brief Description of the Drawings

These and other objects, features and advantages of the present invention will be apparent from the following detailed description and the appended claims and drawings in which:

FIG. 1 is a block diagram of a combined video and graphics processing system; and

FIG. 2 is a block diagram of a BME of FIG. 1.

Detailed Description of the Preferred Embodiments

Referring to FIG. 1, a combined video and graphics processing system 10 is shown. The system 10 generally implements a bus 12 accessed by a graphics CPU 14 and a data processor 100. In one example, the data processor 100 may be implemented as a block move engine (BME). In another example, the data processor 100 may be implemented as a block modify and move engine (BMME) 100. The BMME 100 may be configured to read data from and write data to the memory 16 via the bus 12. A video capture block 20 may also interface with the memory 16 via the bus 12. The memory 16 may also be common to the graphics CPU 14. The processing system 10 may also have a display driver 18 for driving a display 400 (e.g., a video display unit (VDU)).

The graphics CPU 14 may be configured to draw graphics objects within the system memory 16. The video capture block 20 may be configured to write moving video or individual stills to the system memory 16. The BMME 100 may be configured to read any graphics or video objects from the memory 16. The BMME 100 may also be configured to manipulate and combine (e.g., composite) the objects and write the result back to the memory 16.

Referring to FIG. 2, a detailed block diagram of the BMME 100 is shown. The BMME 100 may have direct read/write access to data in the system memory 16 via the bus 12. Data in the system memory 16 may be stored in any format suitable for graphics or still images. Such data formats include bitmaps, color look-up table (CLUT) idiocies, YUV (with or without alpha), RGB (with or without alpha) and alpha planes. Video data formats typically, however, not necessarily, comprise Y, Cb and Cr samples which may be stored interleaved or in separate "frame stores". The sampling structure for the chrominance components may be similar to the luminance components or there may be fewer samples either horizontally or vertically. However, other appropriate data formats may be implemented in order to meet the design criteria of a particular implementation.

The BMME 100 generally comprises a data modification block (or circuit) 102, a composite block (or circuit) 104, a bit expansion block (or circuit) 106, a bit expansion block (or circuit) 108, a bit expansion block (or circuit) 110, a buffer 112, a buffer 114, a buffer 116, a bit reduce block (or circuit) 118 and a buffer 120. The buffers 112, 114 and 116 may directly interface with the system memory 16 via the bus 12. The buffers 112, 114 and 116 may be coupled to the bit expand circuits 106, 108 and 110, respectively. The bit expand circuit 106 may generate a signal (e.g., FRONT) that may be presented to the data modification circuit 102. The bit expand circuit 108 may generate a signal (e.g., BACK) that may be presented to the composite block 104. The bit expand circuit 110 may generate one or more signals (e.g., MASK and/or ALPHA).

The data modification circuit 102 may generate a signal (e.g., FRONT') in response to the signal FRONT. The signal FRONT' may be presented to the composite block 104. The composite block 104 may generate a signal (e.g., RESULT) in response to the signals FRONT', BACK, MASK and/or ALPHA. The signal RESULT may be presented to the bit reduce circuit 118. An output of the bit reduce circuit 118 may be presented to the buffer 120. The buffer

120 may be configured to interface with the memory 16 via the bus 12.

The data stream FRONT may be data which represents the graphic or video foreground. The data stream BACK may be data which represents the graphic or video background. The data stream MASK and/or ALPHA may control the switching of the picture between foreground and background. The bit expand blocks 106, 108 and 110 may be configured to generate the signals FRONT, BACK, MASK and/or ALPHA in response to data from the system memory 16. The BMME 100 may provide an output to the system memory 16 via the bit reduce block 118 and the buffer 120.

The BMME 100 may operate on regions of data stored in the memory 16 which may be in graphics or video data formats. The buffers on the input data paths (e.g., 112, 114, 116) and the output data paths (e.g., 120) of the BMME 100 may incorporate memory address generators to ensure that data is read from or written to the system memory 16 in the desired order. Furthermore, the BMME 100 may not have to scan data in the system memory 16 as a raster scan (e.g., starting from the top left of a region). Scanning may proceed in any direction. The region covered may not need to be rectangular, if the address generators in the buffers 112, 114, 116, and 120 are sufficiently programmable to allow

regions of other shapes (e.g., triangular, etc.). In addition, the directions in which the memory 16 is read from and written to by the BMME 100 may differ for various data paths through the BMME 100. Therefore, the BMME 100 may provide the effect of reversing one or more objects left to right or top to bottom before compositing within the composite 104.

The bit expand blocks 106, 108 and 110 may be optionally implemented to expand source data read from the system memory 16 to a common format, when necessary. The bit expand blocks 106, 108 and 110 may enable the data to be combined easily in the composite block 104. In a first example, data in a 16-bit RGB format (e.g., RGB565) may be expanded to 24-bit RGB format (e.g., RGB888) by adding extra LSBs to each color component of the data. In a second example, CLUT index format data may be expanded to a true-color format by alternatively introducing constant "fill" colors on any data input in the bit expand blocks 106, 108 and 110.

In addition to being modified in the bit expand block 106 the data FRONT may be modified in the data modification block 102 to ensure a format which matches the data BACK and the output data RESULT. The data modification block 102 may perform data modification that may comprise:

performing conversion between video and graphics formats, including changes to the luminance/chrominance sampling structure;

combining (e.g., interleaving) (or separating) particular color components;

scaling and/or filtering of the image, if a size or image bandwidth mismatch;

changing the color space (RGB<=>YUV);

making gain and/or level adjustment to the color components; and/or

performing gamma correction.

The modified data FRONT' may then combine with the data BACK within the composite circuit 104 to generate the image data RESULT. The composite block 104 may be configured to perform bitwise logical operations and/or alpha blending. The composite block 104 may be controlled by the value MASK, the value ALPHA or a "transparency-awareness" value derived from the data value FRONT.

The bit reduce block 118 may provide a way of reducing the number of bits per pixel in the output data RESULT. The bit reduce circuit 118 may reduce a common format generated by the composite block 104 to a smaller format, when necessary (e.g., for storage in the system memory 16).

Although the BMME 100 illustrates a data modification block 102 in the data path FRONT, similar data modification functions may be included in data paths BACK, MASK and/or ALPHA. The BMME 100 may be configured to read any graphics or video objects. The BMME 100 may also be configured to manipulate and combine video objects. The BMME 100 may be configured to write the results back to a system memory.

While the invention has been particularly shown and described with reference to the preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the spirit and scope of the invention.